A Fast Fractal Coding Method for Image with Primary Additional Errors

Shuai Liu
College of Computer Science, Inner Mongolia University, Hohhot, China
School of Physical Science and Technology, Inner Mongolia University, Hohhot, China
*Corresponding author, Email: cs_liushuai@imu.edu.cn

Mengxi Liu, Qi Jia, Lingyun Qi, and Haipeng Li
College of Computer Science, Inner Mongolia University, Hohhot, China
Email: wn_fu@sohu.com

***Abstract*—Today, in the multimedia encoding technology, fractal image coding is an effective coding method without resolution. The effectiveness is because of the high compressing ratio of fractal image coding. But the computational complexity of this coding method is so high that it needs long encoding time. In this paper, a novel fast fractal coding method is constructed to decrease the coding time by the capture of primary additional error values. This method is a universal algorithm, which is independent of image types. First, we abstract the additional error values from classic image coding. Then, we present a method to abstract the primary error values with a given rule of weight. Moreover, the encoding and decoding processes are reformed to store the primary additional error values. Finally, experimental results shows the improved fractal image coding method has higher compressing ratio and better effectiveness (signal to noise ratio) than the classic algorithm.**

***Index Terms*—Fractal Coding; Image Coding; Primary Additional Error; Compressing Ratio; Signal to Noise Ratio**

I. INTRODUCTION

Today, multimedia is used everywhere in the human society. But one bottleneck of multimedia improvement is that its large size needs more space to apply. In order to decrease the size of multimedia, many encoding method are presented. In this way, image coding, which is basis of multimedia coding, is a highlight in this domain.

Nowadays, there are many image encoding methods, such as discrete cosine transform (DCT) [1], Huffman code [2], wavelet image coding [3], etc. Also, there are many international standards, which have been presented by these coding methods, such as BIG, JPEG, H.263, MPEG, etc.

However, the basic ideas of these image coding methods are similar so that the compressing ratios are also similarly [4, 5]. So we need a novel thinking to code images with higher compressing ratio because of the larger images. In this way, fractal image coding is created by the self-similar in nature.

In real world, the geometrical form can be classified as two kinds. One is regular and smooth, which can be described by traditional geometry. Contrarily, the other is rough and anomalistic, which can't be described by traditional geometry. Besides, the natural objects usually have rough and anomalistic forms. So a novel subject is created to research in this domain, which is called fractal geometry [6, 7]. Admittedly, there are much natural scenery is fractal, coastline, mountain's shape, stream, tree, lightning, etc. Nowadays, when fractal theory is combined with computer technology, it becomes an interdisciplinary and nonlinear subject. Fractal images coding is such a technology in this subject, which depends the fractal geometrical form in the images.

Fractal image coding technology is based on the local self-similar of natural images. It uses contractive affine transformation (CAT) to iterate a created image to the coded image. In fractal coding technology, we only need to store the quantization parameters of CAT, whose size is much smaller than the original image. In this case, it can reach the image compression with higher compressing ratio.

After encoding, decoding is also a novel fast iterating process. The existence and uniqueness of the iteration are proved by Banach fixed point theorem. It means that we can use any image to iterate the original image, and the original image is limit of the iteration. So the main problem in fractal coding is to find the best approximation of CAT.

So we find that the fractal coding is a finite distortion method. It means that we can decrease coding time when we decrease part of quality of the coding image. However, this method can be applied when the real-time is required without quality. But in many domains, both real-time and qualities of images are required. So researchers have to find faster fractal coding methods.

In this paper, we decrease coding time by used larger element of the coding set. In other word, the search space changes to small. In additional, we exact primary errors to store in iterating space in order to compensate decoding image. In this case, the fast fractal coding method both has better coding time and coding quality.

The remainder of the paper is organized as follows. We achieve the additional error value from classic image

doi:10.4304/jmm.9.7.955-962

coding, and abstract the primary error value in Section 2. Then, we present the reformed coding process and structure in Section 3. It needs to be reformed because of the storage of primary errors. Moreover, experimental results are presented and analyzed in Section 4. Finally, Section 5 summarizes the whole paper.

II. Related Works

First, in year 1988, Barnsley and Jacquin used iterated function system (IFS) and Recurrent Iterated Function System (RIFS) to code some images [8]. They found that the highest compressing ratio is more than 10000:1. In year 1992, Jacquin achieved self-adaptive fractal coding method in computer [9]. This is the symbol of fractal coding, and means the generation of fractal coding. Meantime, Monro and Dudbridge presented another fractal coding method by used fractal block [10]. Then, Bedford et al extended research of Jacquin, and presented a fractal coding method for monochrome images [11]. Later, Kim and Park presented a coding method of still image [12]. They reach the method by fractal approximation of the image. Kim et al used fractal coding into video sequence [13]. Their research focused on the mapping and non-contractive interframe mapping in the video. Chang and Kuo presented an iteration-free fractal image coding method [14]. Their work is based on the designed domain pool.

After year 2000, due to the requirement of multimedia, especially the images, fractal image coding developed more rapidly than before. Many researchers studied many methods to increase the coding rate and decrease the coding time. In 2002, Li et al used fuzzy image metric into fractal coding [15]. Lai etc presented a fast fractal image coding with kick-out and zero contrast conditions [16]. Belloulata used a non-iterative block clustering to code subbands [17]. Wang etc researched into no-search fractal image coding with a modified gray level transform and fitting plane [18, 19]. His team also paid attention to fractal coding with wavelet transform [20-21]. Meanwhile, Lu etc studied Huber fractal coding with fitting plane [22]. Bhayani and Thanushkodi compared fractal coding methods of medical image compression [23]. Later, there were also some coding methods with fractals. Our team reached some results in this area. We researched fractal properties in k-M set and use it into facial capture [24-25].

Fractal image coding has many advantages, such as high compressing ratio and independence with resolution. However, the coding has high complexity of computation and long coding time. It is because fractal coding needs to find the best matching block for every input sub-block in a large matching set. Admittedly, the result of best matching block can be found by global search of whole matching set. But the computational cost is too high to apply. Though there are many methods are presented to solve this problem, these methods are all for special application (type of image). So in this paper, a universal method is presented to fit all image types when the image can be transformed to a matrix.

III. Primary Additional Error Value

First, we present the steps of a fractal image coding method in Fig. 1.

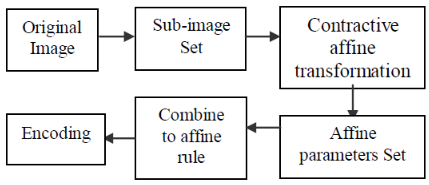


Figure 1. Processes of fractal image coding

Then, we present fractal image coding method step by step in Alg. 1.

Algorithm 1. Fractal Coding Method

Input. Original Image $I(n\times n)$, where n is the size of image I; c is the size of each element in coding set $(c|n)$; f is the size of each element in affine set $(f < c)$.

Output. Encoding file F.

Step 1.

To find a partition I_r of I ($r = 1 \ldots d, d = \left(\frac{n}{c}\right)^2$), where $\forall$ i, j|i $\neq$ j $\rightarrow$ $I_i \cap I_j = \emptyset$ and $\cup_{1\le i\le d} I_i = I$ are all true. The size of I_r is c.

Step 2.

To divide I to D_k. The size of D_k is f.

Step 3.

To select the best affine transformation from all D_k for each I_r. Then, to store the affine transforming table as encoding file to output.

Alg. 1 finished.

In order to corresponding to Alg. 1, we have decoding method in Alg. 2.

Algorithm 2. Fractal Decoding Method

Input. Encoding file F, which contains s, o, (x, y), direction where s is scaling of luminance, o is offset of luminance, (x, y) is affine starting position, direction has eight values (1-8) and denotes the types of equilong transformations; iteration time T; c and f are same to Alg.1.

Output. Decoding image D.

Step 1.

For each rectangular area Di with size f as a decoding area, to get the corresponding s and o as the affine transforming parameters of this area.

Step 2.

For every decoding area Di, to get random image Ri with size c as affine image, then affine mapping it to an image with size f by corresponding parameters s and o.

Step 3.

To collage all rectangular areas to an image D'. Then, let R_i = corresponding part of D', iterating step 1-3 until iterating time =T.

Step 4.

To output D' as D.

Alg. 2 finished.

Admittedly, there exist errors between D and I for any affine transformation. It is because that Eq. 2 is applied to instead of Eq. 1 in real application.

$$\frac{1}{B^2}\min_j\{\min_{s,o\in R,|s|<1}\left\|R_i - \left(s\cdot D_j + o\cdot E\right)\right\|^2\} \qquad (1)$$

$$\min_j\{\min_{s,o\in R}\|R_i - (s\cdot D_j + o\cdot 1)\|\} = \min_{s,o\in R}\left\|R_i - \left(s_i\cdot D_{m_i} + o_i\cdot E\right)\right\| \qquad (2)$$

In Eqs. 1-2, B is the grey level, E is the identity matrix, || || is the vector norm which is usually 2-norm in fractal coding. R_i is each affine sub- image, D_j is each pattern sub-image, mi is serial number of the best D_j, si is scaling of R_i and o_i is offset of R_i in affine mapping.

In this way, we can extract additional error $\Delta_{n\times n}$ from Eq. 3.

$$\Delta_{n\times n} = I_{n\times n} - \bigcup_{1\le j\le d} D_j \tag{3}$$

For a general nature image, the value zero in Δ is usually a little. But many values are small. Realistically, to ensure the quality of the image by visual angle, we don't interest in those small values in Δ. Meanwhile, in order to drop the blocking effect in the decoding image, we have to store those error values Δp and Δq that Δp locates the edge e_p of D_p, Δq locates the edge e_q of D_q and e_p=e_q. In this case, we have an evaluation standard to extract the valuable errors from Eq. 4. Then we use and named them "additional error values".

$$w_{i,j} = \Delta_{i,j} + \sum_{k=1}^{5} u_k \cdot \frac{1}{k-1} \sum_{t_1+t_2=k, t_1,t_2\ge 0} \left(\left|w_{i,j} - w_{i-t_1, j+t_2}\right|\right) \tag{4}$$

In Eq. 4, u_k is penalty point which is smaller when k is larger. In our method, we don't think that there is blocking effect when k>5. Then, with the sequence of all w_{ij} by their values, we extract the primary additional error values. The number of the primary additional error values is d/2, which can be stored in the encoding file with same size. Then, we present a fast fractal coding method in the following Section.

IV. A Fast Fractal Coding Method with Primary Additional Error Value

Since we have extracted the additional error values, the quality of the decoding image is high enough. However, the coding time is still large. So we can use a higher R and D with same c/f to decrease coding time. Then we can compare the fast coding method to the classic one.

Assuming that variable parameters of the classic method are c and f for a static image with n size, we use c1=2c and f1=2f as the parameters of ours. Then, we can compute the compressing ratio of the two methods by Eqs. 5-6, and the coding time of the two methods by Eqs. 7-8 when assuming the time of one affine mapping is t.

In these equations,

$$CR_{classic} = \frac{n^2}{6\cdot\left(\frac{n}{f}\right)^2} = \frac{f^2}{6} \tag{5}$$

$$CR_{fast} = \frac{n^2}{6\cdot\left(\frac{n}{2f}\right)^2 + 8d^2} = \frac{f^2}{8} = \frac{3}{4} CR_{classic} \tag{6}$$

$$T_{classic} = t\cdot\left(\frac{n}{c}\right)^2\left(\frac{n}{f}\right)^2 \tag{7}$$

$$T_{fast} = t\cdot\left(\frac{n}{2c}\right)^2\left(\frac{n}{2f}\right)^2 = \frac{1}{16} T_{classic} \tag{8}$$

In this case, we know that we have both higher compressing ratio and smaller compressing time with the improved algorithm. Then, we present the encoding and decoding algorithms in Algs. 3-4.

Algorithm 3. Fast Fractal Coding Method
Input. Same to Alg. 1.
Output. Same to Alg.1
Step 1.
Encoding F by Alg.1 with parameters c_1 and f_1.
Step 2.
Extracting primary additional error values with Eq. 4 for Δ. The number of primary additional errors is $8d^2$. Its coding rule likes $P = \{p_{xy}\}$, where (x,y) is the position of errors and pxy is the value of the error.
Step 3.
Attaching $V = \{v_{xy}\}$ to F where $v_{xy} = x\cdot 2^{16} + y\cdot 2^8 + p_{xy}$ after rewrite original position (e_x, e_y) to $e = e_x\cdot 2^8 + e_y$. Then, F with addition V is the output.
Alg. 3 finished.

Algorithm 4. Fast Fractal Decoding Method
Input. F with additional V.
Output. Same to Alg. 2.
Step 1.
Extracting $P = \{p_{xy}\}$ where $p_{xy} = v_{xy} \bmod 2^8$, $y = \left(\frac{v_{xy}-p_{xy}}{2^8}\right) \bmod 2^8, x = \frac{v_{xy}-p_{xy}-y\cdot 2^8}{2^{16}}$. Extracting (ex, ey) with $e_y = e \bmod 2^8$ and $e_x = \frac{e-e_y}{2^8}$.
Step 2.
Decoding D with same process of Alg. 2.
Step 3.
Add P to D by used $d_{xy} = d_{xy} + p_{xy}$. D is the output.
Alg. 4 finished.

Because the scale of F is $F_c = (n/2c)^2 = d/4$, we have that $d/2 = 2F_c$. It means we have two additional dimensions to attach these values. In additional, we rewrite (e_x, e_y) to e, which economizes one dimension. Thus, we drop value of the 6th dimension. So the total scale doesn't increase.

Furthermore, in the decoding period, we spend additional time to extract V and e. But these are only basic computation, which spends only a little computational time.

V. Experiment and Analysis

In this paper, we use four images as the examples. In these examples, the 1st is classic "Lena", the 2nd is named "building", the 3rd is named "bride", and the 4th is named "windmill". They are all reformed to greyscale images with size 256×256 and presented in Fig. 2. In Fig. 2, we name these figures in following. Fig. 2a is named as "Lena", Fig. 2b is named as "building", Fig. 2c is named as "bride", and Fig. 2d is named as "windmill".

Figure 2. Original images

a b

Figure 3. Decoding Images with both classic and improved method in image “Lena” (by six iterated time)

a b

Figure 4. Decoding Images with both classic and improved method in image “building”

In our experiments, c=8 and f=4. So c_1=16 and f_1=8. Figs. 3-6 show the comparisons of the decoding images of these two methods. Fig. 3 shows the comparison of the decoding images of image “Lena”. Fig. 4 shows the comparison of image “building”. Fig. 5 shows the comparison of image “bride”. Fig. 6 shows the comparison of image “windmill”. In the sub-figure “a” of each figure, we decode the image by the classic method. In the sub-figure “b” of each figure, we decode the image by the improved method.

a b

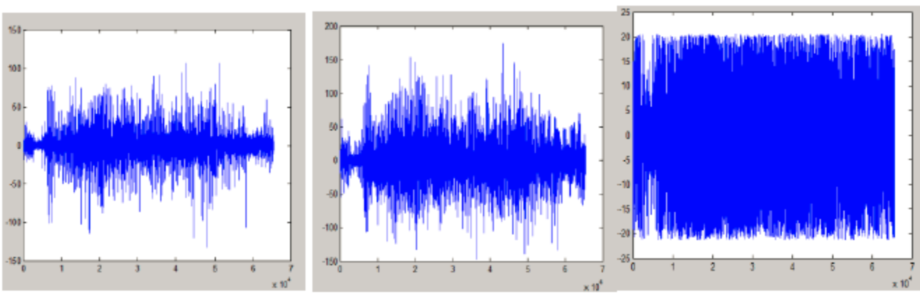
Figure 5. Decoding Images with both classic and improved method in image “bride” (by six iterated time)

a b

Figure 6. Decoding Images with both classic and improved method in image “windmill” (by six iterated time)

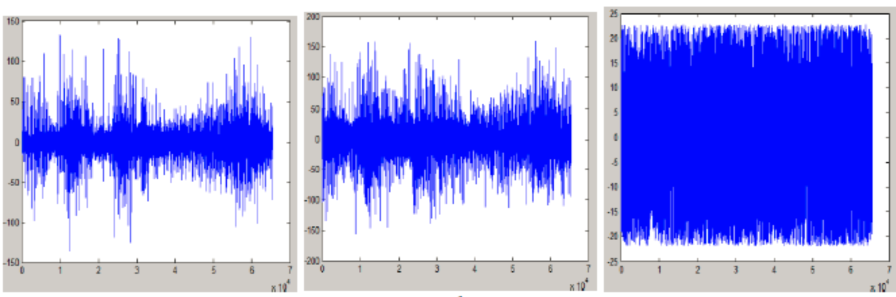
In fact, we can’t judge which decoding image is better by used human vision. So we compute PSNR to measure the quality of the two decoding image from Eq. 9.

$$\mathrm{PSNR} = 10\cdot\log_{10}\frac{n^2\cdot 255^2}{\sum_{i=1}^{n^2}(I_i-D_i)^2} \tag{9}$$



a b c

Figure 7. Comparison of Additional error values with both the classic, improved method (without primary additional error values), and improved method (with primary additional error values) in the image “Lena”



a b c

Figure 8. Comparison of Additional error values with both the classic, improved method (without primary additional error values), and improved method (with primary additional error values) in the image “building”

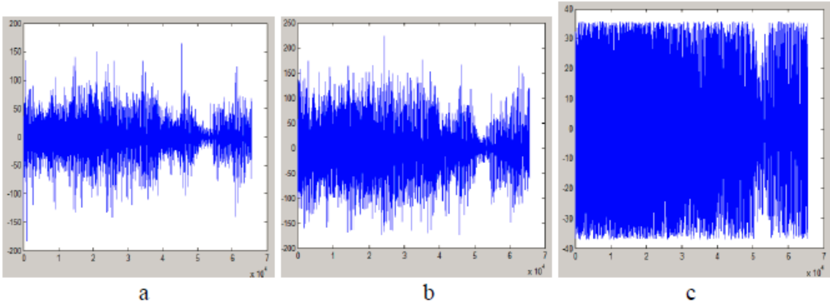


Figure 9. Comparison of Additional error values with both the classic, improved method (without primary additional error values), and improved method (with primary additional error values) in the image “bride”

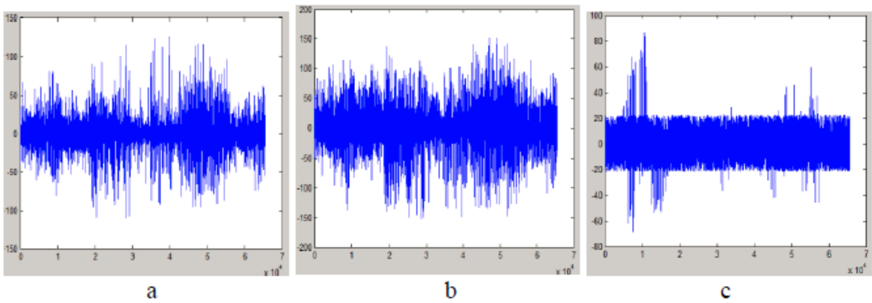


Figure 10. Comparison of Additional error values with both the classic, improved method (without primary additional error values), and improved method (with primary additional error values) in the image “windmill”

Figure 11. Decoding images with each iterated time (1-5) for image “Lena”

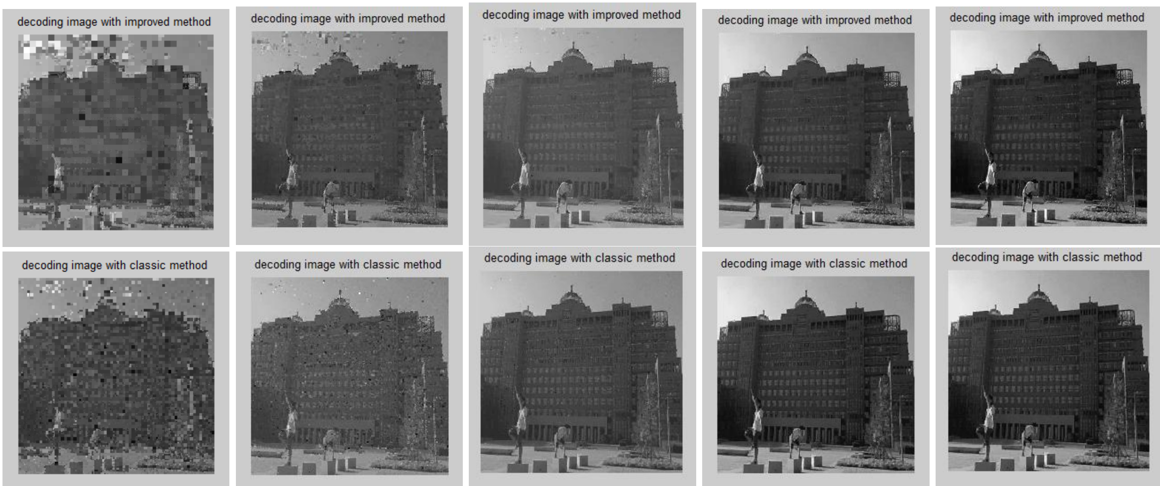


Figure 12. Decoding images with each iterated time (1-5) for image “building”

Then, we provide Figs. 7-10 to show the comparisons of additional error values between the classic method and the improved method in these four images. Fig. 7 shows the comparison of the additional error values with these two methods in image "Lena". Fig. 8 shows the comparison of image "building". Fig. 9 shows the comparison of image "bride". Fig. 10 shows the comparison of image "windmill".

Similar to Figs. 3-6, in Figs. 7-10, the sub-figure "a" of each figure shows the additional error values with the classic method, the sub-figure "b" of each figure shows the additional error values of the improved method without primary additional error values, and the sub-figure "c" of each figure shows the additional error values of the improved method with primary additional error values. First, we know that the sizes of the matrix are same because the number of points is 256×256=65536.

In these figures, we have that the additional error values of classic method are lower than the improved method without primary additional error values. It is because that the size of divided blocks in classic method is smaller than the size in improved method (the mean of value is 50~80 and maximum value is 150~200 in the classic method, the mean of value is 80~100 and maximum value is 200~250 in the improved method without primary additional error values).

TABLE I. EXPERIMENTAL RESULTS OF CLASSIC FRACTAL CODING

Image	Coding Time (s)	Decoding Time (s)	PSNR (dB)	Compressing ratio
Lena	1003.496	1.327	30.0933	2.67
Building	1212.999	1.032	29.0667	2.67
Bride	1228.219	1.014	25.9097	2.67
Windmill	1235.142	1.402	28.7373	2.67

However, we also find that the additional error values change to 25~40 per point when the primary additional error values are added. It means that the primary additional error values increase the compressing effect indeed.

TABLE II. EXPERIMENTAL RESULTS OF FAST FRACTAL CODING

Image	Coding Time (s)	Decoding Time (s)	PSNR (dB)	Compressing ratio
Lena	74.134	0.799	31.5946	3.59
Building	80.497	0.973	30.4684	3.59
Bride	81.135	0.924	26.8855	3.59
Windmill	81.775	0.856	31.7594	3.59

Figure 13. Decoding images with each iterated time (1-5) for image "bride"

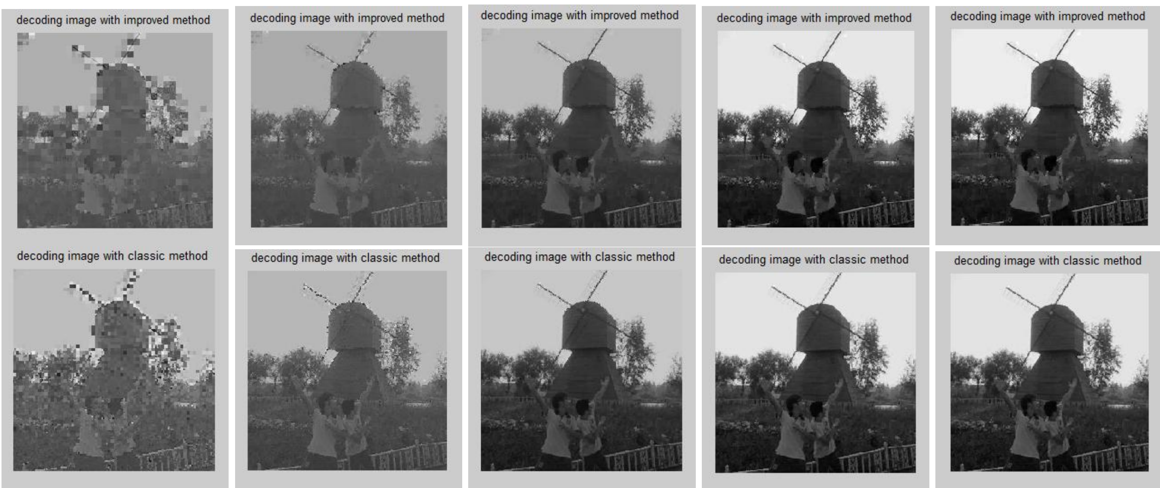


Figure 14. Decoding images with each iterated time (1-5) for image "windmill"

In additional, we have the results of the classic and improved algorithms in tables 1-2. The two tables show the coding time, decoding time, PSNR and compressing ratio of classic algorithm of these two algorithms. In table 1, we find that the coding time is very long (>1000s) when it reaches the well PSNR (>25). Though it has short decoding time, it can not be used in the real world because of the long coding waste. But in table 2, we find both the encoding and decoding time are all well enough to use in application.

Then, we compute the ratio of the two methods and find they are all nearby 16. Concretely, ratio of Lena is 13.54 and others are all between 15 and 16. It validates our conclusion of coding time in Eqs. 5-8. Furthermore, the ratio of “compressing ratio” of the two is 0.744. It also validates our conclusion of compressing ratio in Eqs. 5-8.

In additional, we show the decoding image of the four original images for each iterated time 1-5 in Figs. 11-14. The upper sub-images of each figure are all decoding with the improved algorithm, and the lower sub-images are all with the classic algorithm. In these figures, we have that the decoding image of improved algorithm is better than the classic algorithm when the iterated time is more than three.

VI. CONCLUSION

In this paper, we present a fast fractal coding method with primary component analysis of additional error value. We extract the primary additional error value and store them into encoding file. We also improve structure of the encoding file in order to store the error value. Moreover, we change the decoding rule of the improved coding method to add the error values to the decoding image. Finally, we use some images to experiment. The experimental results show that the fast fractal coding has higher coding speed and better “signal to noise ratio”.

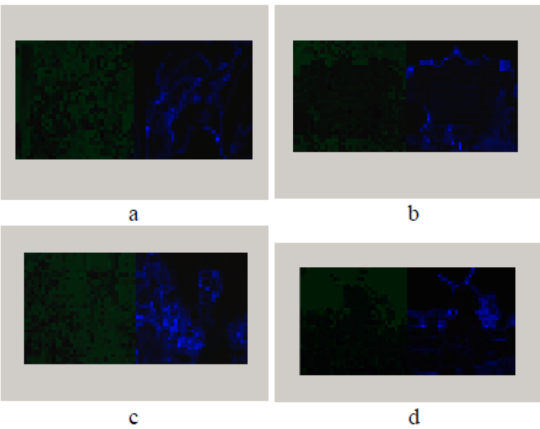


Figure 15. The encoding images of the four original images

In future, our work will pay attention to microcosmic coding rule. We project that we will construct a novel storage rule to decrease space of additional error value. We also project that we will construct a rule of indemnity by fuzzy set. Another area we want to study is to use parallel environment to increase coding time [26-28].

Another work will be the encryption with fractal coding. In Figs. 15, we have presented the encoded images of these four images. Fig. 15a is corresponding to “Lena”, Fig. 15b is corresponding to “building”, Fig. 15c is corresponding to “bride”, and Fig. 15d is corresponding to “windmill”. In Fig. 15, we find that the encoding image is so confusing to the original image. So we will use this technology into encryption in future.

ACKNOWLEDGMENT

This work is supported by Grants Programs of Higher-level talents of Inner Mongolia University [No. 125126, 115117], Innovation Training Program of Graduates of Inner Mongolia University [201315226], Scientific projects of higher school of Inner Mongolia [No. NJZY13004], National Natural Science Foundation of China [No.61261019, 61262082].

The authors wish to thank the anonymous reviewers for their helpful comments in reviewing this paper.

REFERENCES

[1] Rao K R, Yip P. Discrete cosine transform: algorithms, advantages, applications. *Academic press, Boston,* 1990.
[2] Huffman D A. A method for the construction of minimum redundancy codes. *Proceedings of the IRE (IEEE),* 40(9) (1952) 1098-1101.
[3] Shapiro J M. Embedded image coding using zerotrees of wavelet coefficients. Signal Processing, *IEEE Transactions on,* 41(12) (1993) 3445-3462.
[4] M. Egmont-Petersena, D. de Ridderb and H. Handelsc, Image processing with neural networks -a review, *Pattern Recognition* 35 (2002) 2279–2301
[5] M. Vetterli, Wavelets and Filter Banks: Theory and Design, IEEE Trans. *Signal Processing,* 40(9)(1992) 2207-2232
[6] Mandelbrot BB, The Fractal Geometry of Nature, *Freeman W H, San Francisco,* 1982.
[7] K Falconer. Fractal Geometry: Mathematical Foundations and Applications, *Second Edition. John Wiley @ Sons, Inc,* 2003.
[8] M. F. Barnsley and A. E. Jacquin, Application of recurrent iterated function systems to images, in Proceedings of the SPIE, *Visual Communications and Image Processing,* 1001 (1988) 122-131
[9] A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Trans. Image Process.* 1 (1) (1992) 18-30
[10] Monro D M, Dudbridge F. Fractal block coding of images. *Electronics letters,* 28(11) (1992) 1053-1055.
[11] Bedford T, Dekking F M, Breeuwer M, etc. Fractal coding of monochrome images. Signal processing: *Image communication,* 6(5) (1994) 405-419.
[12] Kim I K, Park R H. Still image coding based on vector quantization and fractal approximation. *Image Processing, IEEE Transactions on,* 15(4) (1996) 587-597.
[13] Kim C S, Kim R C, Lee S U. Fractal coding of video sequence using circular prediction mapping and noncontractive interframe mapping. *Image Processing, IEEE Transactions on,* 7(4) (1998) 601-605.
[14] Chang H T, Kuo C J. Iteration-free fractal image coding based on efficient domain pool design. *Image Processing, IEEE Transactions on,* 9(3) (2000) 329-339.
[15] Li J, Chen G, Chi Z. A fuzzy image metric with application to fractal coding. Image Processing, *IEEE Transactions on,* 11(6) (2002) 636-643.

[16] Lai C M, Lam K M, Siu W C. A fast fractal image coding based on kick-out and zero contrast conditions. *Image Processing, IEEE Transactions on,* 12(11) (2003) 1398-1403.
[17] Belloulata K. Fast fractal coding of subbands using a non-iterative block clustering. *Journal of Visual Communication and Image Representation,* 16(1) (2005) 55-67.
[18] Wang X Y, Wang S G. An improved no-search fractal image coding method based on a modified gray-level transform. *Computers & Graphics,* 32(4) (2008) 445-450.
[19] Wang X Y, Wang Y X, Yun J J. An improved no-search fractal image coding method based on a fitting plane. *Image and Vision Computing,* 28(8) (2010) 1303-1308.
[20] Zhang Y, Wang X. Fractal compression coding based on wavelet transform with diamond search. *Nonlinear Analysis: Real World Applications,* 13(1) (2012) 106-112.
[21] Wang X Y, Zhang D D. Discrete wavelet transform-based simple range classification strategies for fractal image coding. *Nonlinear Dynamics,* 75(3) (2014) 439-448.
[22] Lu J, Ye Z, Zou Y. Huber fractal image coding based on a fitting plane. *Image Processing, IEEE Transactions on,* 22(1) (2013) 134-145.
[23] Bhavani S, Thanushkodi K G. Comparison of fractal coding methods for medical image compression. *IET Image Processing,* 7(7) (2013) 686-693.
[24] Liu S, Cheng X, Lan C, etc, Fractal property of generalized M-set with rational number exponent. *Applied Mathematics and Computation,* 220 (2013), 668-675.
[25] S Liu, W Fu, W Zhao, etc. A Novel Fusion Method by Static and Moving Facial Capture. *Mathematical Problems in Engineering,* (2013) doi:10.1155/2013/503924.
[26] Liu M, Liu S, Fu W, etc, Distributional Escape Time Algorithm based on Generalized Fractal Sets in Cloud Environment. *Chinese Journal of Electronics* (In press).
[27] S Liu, W Fu, H Deng, etc. Distributional Fractal Creating Algorithm in Parallel Environment, *International Journal of Distributed Sensor Networks,* (2013) doi:/10.1155/2013/281707.
[28] G Yang, S Liu, "Distributed Cooperative Algorithm for Set with Negative Integer by Fractal Symmetrical Property," *International Journal of Distributed Sensor Networks,* (2014) doi:10.1155/2014/398583.